The thing the time that the ij, N



4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

CLAIMS

1. In a paging operating system having physical memory for holding information and secondary storage comprising a page file for receiving information that is paged out from the physical memory, a computer-implemented method of protecting information comprising:

encrypting information using a key that is page-locked in the physical memory; and

paging out, to the page file, the encrypted information.

- 2. The computer-implemented method of claim 1 further comprising prior to said encrypting, creating the key and page locking the key in the physical memory.
- 3. The computer-implemented method of claim 2, wherein said creating the key comprises creating the key during system boot up.
- The computer-implemented method of claim 2, wherein said creating the key comprises generating a random key with a random key generator.
- The computer-implemented method of claim 4, wherein said generating comprises using RSA RC4 as an encryption algorithm to generate the key/.



6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

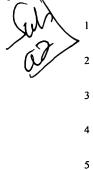
The computer-implemented method of claim 1, wherein said 6. encrypting comprises:

calling an operating system kernel; the kernel using the page-locked key to encrypt the information.

- The computer-implemented method of claim 6, wherein said calling 7. is performed by an application.
- 8. The computer-implemented method of claim 6, wherein said calling is performed by an operating system memory manager.
- One or more computer-feadable media having computer-readable 9. instructions thereon which, when executed by a computer, perform the computerimplemented method of claim 1.
- 10. An operating system programmed with instructions which, when implemented by the operating system, implement the method of claim 1.
- In a paging operating system having main memory for holding 11. information and secondary storage comprising a page file for receiving information that is paged out from the main memory, a computer-implemented method of protecting information comprising:

page-locking a key in main memory;

restricting access to the page-locked key to only the operating system kernel;



calling the operating system kernel to encrypt information;
accessing the page-locked key with the operating system kernel; and
using the operating system kernel to encrypt the information with the pagelocked key.

- 12. The computer-implemented method of claim 11, wherein said calling is performed by an operating system memory manager.
- 13. The computer-implemented method of claim 11, wherein said calling is performed by an application.
- 14. The computer-implemented method of claim 11 further comprising prior to said calling:

designating at least one page in the main memory with a designation; recognizing the designation and, responsive thereto, calling the operating system kernel to encrypt the information.

- 15. The computer-implemented method of claim 14, wherein said recognizing is performed by the memory manager.
- 16. The computer-implemented method of claim 11, wherein said calling comprises specifying a memory location and a memory size associated with the information to be encrypted.

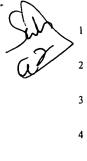


17. One or more computer-readable media having computer-readable instructions thereon which, when executed by a computer, perform the computer-implemented method of claim 11.

- 18. An operating system programmed with instructions which, when implemented by the operating system, implement the method of claim 11.
- 19. In a paging operating system having main memory for holding information and secondary storage comprising a page file for receiving information that is paged out from the main memory, a computer-implemented method of handling encrypted information comprising:

accessing encrypted information in the page file; and decrypting the encrypted information with a key that is page-locked in the main memory.

- 20. The computer-implemented method of claim 19 further comprising placing the decrypted information in a page of main memory.
- 21. The computer-implemented method of claim 19 further comprising placing the decrypted information in a page-locked page of main memory.
- 22. The computer-implemented method of claim 19, wherein the page-locked key is accessible only to the operating system kernel.



23. One or more computer-readable media having computer-readable instructions thereon which, when executed by a computer, perform the computer-implemented method of claim 19.

- 24. An operating system programmed with instructions which, when implemented by the operating system, implement the method of claim 19.
- 25. In a paging operating system having main memory for holding information and secondary storage comprising a page file for receiving information that is paged out from the main memory, a computer-implemented method of protecting information comprising:

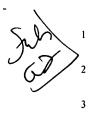
allocating a non-pageable page of main memory;

generating a random key; and

storing the random key in the non-pageable page of main memory, the random key being configured for use by the operating system to encrypt information that might be paged out to the page file.

- 26. The computer-implemented method of claim 25, wherein said generating comprises using an RSA RC4 encryption algorithm.
- 27. The computer-implemented method of claim 25, wherein said allocating takes place during system boot.

Lee & Hayes, PLLC



7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

28. One or more computer-readable media having computer-readable instructions thereon which, when executed by a computer, perform the computer-implemented method of claim 25.

- 29. An operating system programmed with instructions which, when implemented by the operating system, implement the method of claim 25.
- 30. In an operating system having main memory for holding information and secondary storage for receiving information that is transferred out of main memory, a computer-implemented method of protecting information comprising:

generating at least one random key by using a random key generation process;

encrypting at least one selected block of information in the main memory with a software component that uses the at least one random key for encryption;

transferring the one encrypted block of information to the secondary storage;

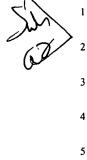
decrypting the one encrypted block of information with the software component that uses the at least one random key for decryption; and

placing the decrypted block of information in the main memory.

31. The computer-implemented method of claim 30, wherein said generating is performed during system boot up.

222324

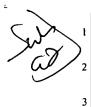
25



32. The computer-implemented method of claim 30 further comprising restricting access to the at least one random key to only the software component.

- 33. The computer-implemented method of claim 30, wherein the software component comprises the operating system's kernel.
- 34. The computer-implemented method of claim 30 further comprising: storing the at least one random key in the main memory; and locking the at least one random key in the main memory so that it does not get transferred to the second storage.
- 35. An operating system programmed with instructions which, when implemented by the operating system, implement the method of claim 30.
- 36. A system for use in protecting pageable information comprising:
 a memory having pageable and non-pageable pages; and
 at least one key stored in the memory in a non-pageable page, the key being
 configured for use in encrypting pageable information.
- 37. The system of claim 36 further comprising a software component that is configured to access and use said one key to encrypt pageable information.
- 38. The system of claim 37, wherein the one key is accessible only to the software component.

Lce & Hayes, PLLC



6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

The system of claim 37 further comprising at least one application 39. configured to call the software component to encrypt the pageable information.

- The system of claim 37 further comprising a memory manager 40. configured to call the software component to encrypt the pageable information.
- 41. A computer program embodied on one or more computer-readable media, the program comprising:

encrypting information with a key that is page-locked in main memory of a computer;

paging out, to secondary storage,/the encrypted information; accessing the encrypted information in the secondary storage; and decrypting the encrypted information with the key that is page-locked in the main memory.

A programmable computer comprising: 42.

a processor;

main memory for holding information;

secondary storage for receiving information that is temporarily transferred out of the main memory;

the computer being programmed with computer-readable instructions which, when executed by the processor, cause the computer to:

encrypt information that is to be transferred to the secondary storage with a key that is locked in the main memory;

transfer the encrypted information to the secondary storage; and

111 Ŋ



6

7

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

decrypt the encrypted information with a key that is locked in the main memory.

- The programmable computer of claim 42/, wherein the instructions 43. cause the computer to generate the key and lock the key in the main memory.
- The programmable computer of claim 42, wherein the key that is 44. used to encrypt the information is the same/key that is used to decrypt the information.
- The programmable computer of claim 42, further comprising a 45. software component that is programmed to encrypt and decrypt the information.
- 46. The programmable computer of claim 45, wherein the software component comprises the operating system's kernel.
- One or more application programming interfaces embodied on one 47. or more computer-readable media for execution on a computer in conjunction with a paging operating system having main memory for holding information and a page file for receiving information that is paged out from the main memory, comprising:

an interface method for encrypting pageable information with a key that is page-locked in the main memory; and

an interface method for decrypting encrypted information that is contained in the page file.

The state of the s

An application programming interface embodied on a computer-48. readable medium for execution on a computer in conjunction with a paging operating system having main memory for holding information and secondary storage comprising a page file for receiving information that is paged out from the main memory, comprising a method for setting an attribute on a page of main memory, the attribute designating that the page must be encrypted with a key that is page-locked in the main memory prior to the page being paged out to the page file.